

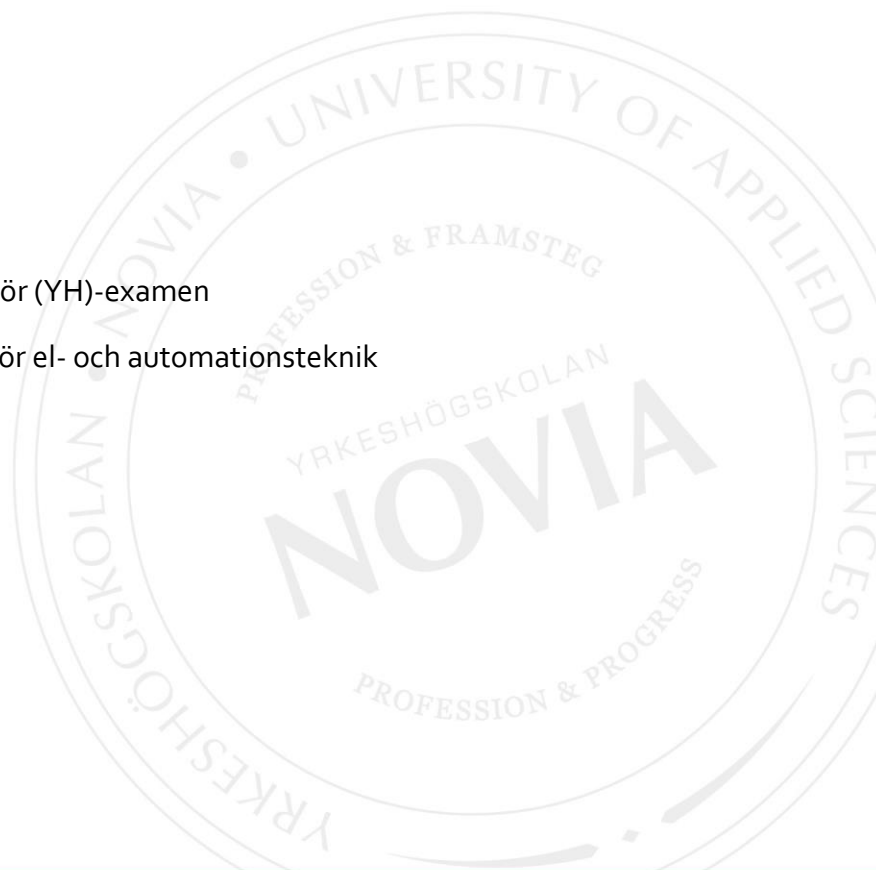
# Integration av tjänsten Unifaun till webbshop

Mikael Norrgård

Examensarbete för ingenjör (YH)-examen

Utbildningsprogrammet för el- och automationsteknik

Vasa 2018



## EXAMENSARBETE

Författare: Mikael Norrgård

Utbildning och ort: El- och automationsteknik, Vasa

Inriktningsalternativ: Informationsteknik

Handledare: Kaj Wikman

Titel: Integration av tjänsten Unifaun till webbshop

---

Datum: 4.9.2018

Sidantal: 20

---

### Abstrakt

Detta examensarbete behandlar utvecklingen av en integration av tjänsten Unifaun till uppdragsgivarens webbshopsmodul. Integrationen ska göra det möjligt att lättare få tillgång till transportdokument och ge webbutikens kunder möjligheten att spåra deras försändelser på ett smidigt sätt. Med hjälp av integrationen kommer de direkt efter att beställningen är gjord få tillgång till spårningsinformation i ett e-postmeddelande. Arbetet utfördes åt företaget Creamarketing Ab Oy.

Målet var att påbörja arbetet med en integration i Silverstripes egen shopmodul Silvershop där kunderna lätt kan följa med var i leveranskedjan deras beställningar befinner sig.

Resultatet blev en produkt som registrerar beställningar i Unifaun Online och vid verkliga försändelser även skickar ut ett e-postmeddelande med spårningsinformation.

---

Språk: Svenska

Nyckelord: Unifaun, webbutik

---

## BACHELOR'S THESIS

Author: Mikael Norrgård

Degree Programme: Electrical Engineering

Specialization: Information Technology

Supervisor: Kaj Wikman

Title: Integration of Unifaun service to webshop

---

Date: September 4, 2018

Number of pages: 20

---

### **Abstract**

This thesis is based on the development of an integration of Unifaun service to the employer's web shop module. The integration enables access to transport documents and give the customers a possibility to track their shipments in an easy way in the webshop. With help from the integration they will get access to tracking information in an email directly after the order is done. The project was created for the company Creamarketing Ab Oy.

The goal was to start the work with an integration in Silverstripe's own shop module Silvershop where the customers easily could follow up where in the delivery chain their orders are.

The result was a product that registers orders in Unifaun Online and also sends an email with tracking information with real shipments.

---

Language: Swedish

Key words: Unifaun, Ecommerce Store

---

# Innehållsförteckning

1	Inledning.....	1
1.1	Uppdragsgivare.....	1
1.2	Syfte .....	1
2	Tekniker.....	2
2.1	Programmeringsbegrepp.....	2
2.1.1	PHP.....	2
2.1.2	Guzzle .....	2
2.1.3	JSON Data.....	4
2.1.4	API .....	4
2.1.5	REST API .....	5
2.2	Ubuntu .....	5
2.3	Unifaun .....	6
2.3.1	Unifaun Online .....	6
2.3.2	Unifaun REST API .....	6
2.4	Silverstripe.....	9
2.5	Apache .....	10
2.6	MariaDB .....	10
2.7	Git .....	10
2.8	Composer.....	10
3	Planering.....	11
3.1	Specifikationer.....	11
3.2	Förberedelse .....	11
4	Utförande.....	12
4.1	Skapande av projektet.....	12
4.2	Guzzle förfrågningar .....	13
5	Resultat och diskussion .....	19
6	Källförteckning.....	20

# 1 Inledning

I detta examensarbete behandlas en integration av tjänsten Unifaun. I examensarbetet skapas också en webbutik i Silverstripe med hjälp av deras egen Silvershopmodul. Silverstripe är ett Content Management System (CMS) med öppen källkod som används till att bland annat uppdatera och skapa nya webbsidor. För mer information om Silverstripe läs kapitel 2.4. Syftet med arbetet är att göra det lättare för uppdragsgivarens kunder att följa med sina beställningar med hjälp av bland annat ett spårningsnummer som kunden erhåller i samband med beställningen.

## 1.1 Uppdragsgivare

Examensarbetet är utvecklat till företaget Creamarketing Ab Oy. Företaget grundades år 1998 i Vasa och skräddarsyr allt från enkla lättanvända hemsidor och webbapplikationer till mycket avancerade tekniska lösningar. Företaget erbjuder till exempel ett verktyg för utrymmesbokningar och info-TV.



Figur 1. Creamarketing logo. (Creamarketing, 2018)

## 1.2 Syfte

Syftet med examensarbetet är att påbörja integreringen av tjänsten Unifaun för att göra det enklare för företagets kunder att på ett smidigt och lättåtkomligt sätt använda sig av Unifauns tjänster i deras webbutiker. Bland annat skulle kunderna enkelt kunna följa med beställningarna i webbutiken med hjälp av ett spårningsnummer istället för att själva gå till leverantörens spårningstjänst och fylla i uppgifterna där för att spåra sitt paket. Kunderna ska nu med hjälp av integrationen av Unifaun kunna följa med var i leveranskedjan deras beställningar befinner sig.

## 2 Tekniker

I detta kapitel presenteras de olika tjänsterna och verktygen som har använts för att utföra examensarbetet. Bland annat Ubuntu för Windows 10, Silverstripe, PHP, Composer och Guzzle.

### 2.1 Programmeringsbegrepp

Det finns många olika begrepp inom programmering. Här nedan beskrivs bara de språk och verktyg som använts i detta examensarbete. Programmeringsspråk som använts förut används kanske inte alls i lika stor utsträckning idag då nya språk har tillkommit som på ett eller annat sätt ersatt de gamla. Datorer utvecklas hela tiden och programmeringsspråken och verktygen likaså.

#### 2.1.1 PHP

PHP är ett scriptspråk med öppen källkod som är vanligt att använda för att skapa bland annat olika applikationer och dynamiska webbplatser. Använder man PHP tillsammans med en databas kan man skapa olika system som används för att dynamiskt ändra, lägga till eller ta bort information. PHP används främst vid skript på serversidan, för att få det att fungera behövs tillgång till en webbservare, en webbläsare och CGI (Common Gateway Interface) som i det här fallet installeras i samband med Apache.

Om man använder sig av en webbservare tillsammans med PHP kan man genom webbservern få tillgång till att se en PHP sida i valfri webbläsare. PHP skript kan också användas vid kommandoradskript och när man skriver skrivbordsapplikationer. Man kan se att ett dokument är skrivet i PHP då det avslutas med ".php". [4][5]

#### 2.1.2 Guzzle

Guzzle är en PHP HTTP klient som gör det möjligt att integrera mot olika webservices med hjälp av att skicka olika HTTP-förfrågningar. Med hjälp av ett enkelt gränssnitt går det till exempel ladda upp JSON-data och göra POST-förfrågningar. I samma gränssnitt går det också att använda sig av både asynkrona och synkrona förfrågningar. Här nedanför finns ett exempel på en Guzzle GET-förfrågan till en API som kräver autentisering med hjälp av ett

användarnamn och lösenord. Som svar fås bland annat statuskoden 200 för en lyckad förfrågan och innehållet i bodyn som en JSON-sträng. [11]

**Kodexempel 1. Exempel på användningen av Guzzle. (Guzzle, 2015)**

```
$client = new GuzzleHttp\Client();
$res = $client->request('GET', 'https://api.github.com/user', [
    'auth' => ['user', 'pass']
]);
echo $res->getStatusCode();
// "200"
echo $res->getHeader('content-type');
// 'application/json; charset=utf8'
echo $res->getBody();
// {"type":"User"...}

// Send an asynchronous request.
$request = new \GuzzleHttp\Psr7\Request('GET', 'http://httpbin.org');
$promise = $client->sendAsync($request)->then(function ($response) {
    echo 'I completed! ' . $response->getBody();
});
$promise->wait();
```

### 2.1.3 JSON Data

JSON eller JavaScript Object Notation är ett datautbytesformat som är enkelt att skriva och läsa för människan. JSON-formatet kan lätt användas till att skicka och ta emot data från en server och är inte beroende av något särskilt programmeringsspråk. [12]

**Kodexempel 2. Exempel på JSON objekt.**

```
"sender":  
{  
  "email": "tester@msn.com",  
  "zipcode": "65100",  
  "name": "Test Sender",  
  "address1": "Testgatan 5",  
  "country": "FI",  
  "city": "Vasa"  
}
```

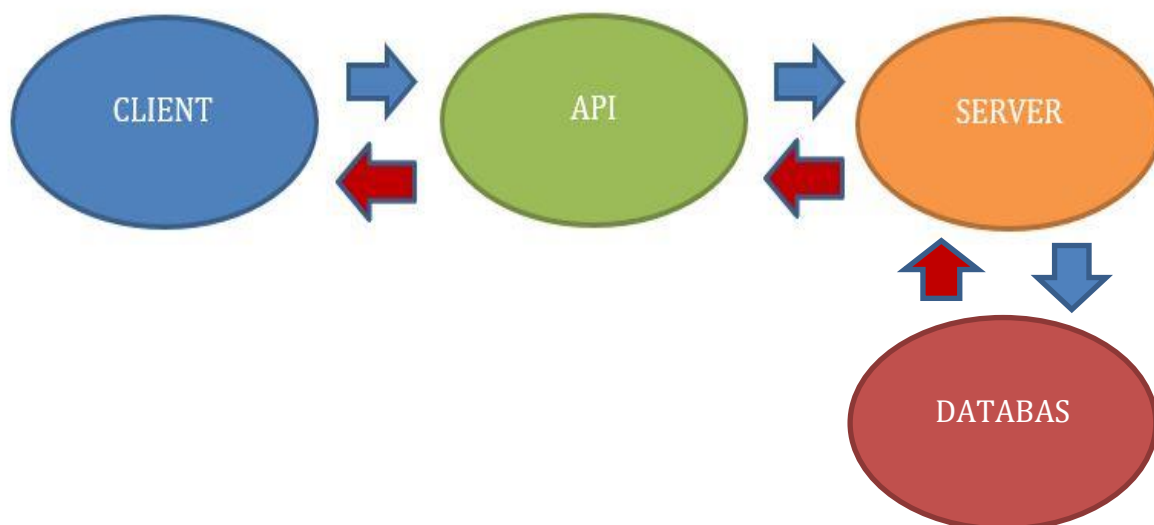
### 2.1.4 API

API eller Application Programming Interface är ett gränssnitt som gör det möjligt för två programvaror att kommunicera med varandra. Vanligtvis är det en server och en klient som kommunicerar med varandra med hjälp av en API. En API kan delvis jämföras med en servitör på en restaurang. Det finns ett antal olika alternativ att välja mellan på en meny och alla har en kort beskrivning vad de innehåller. Ett eller flera alternativ väljs från menyn som restaurangpersonalen senare tillverkar och som servitören levererar till bordet utan att restaurangbesökaren nödvändigtvis vet hur maten har tillretts.

En API kan på liknande vis beskrivas som en lista med olika funktioner som utvecklaren kan välja emellan som senare utförs av gränssnittet. Det viktigaste är att utvecklaren ser att funktionen som behövs finns tillgänglig utan att nödvändigtvis direkt ha full koll på hur den exakt fungerar. En del API:s kräver dock att användaren fyller i data själv för att få ett önskat resultat och därför är inte restaurangexemplet en helt korrekt beskrivning för en API men det ger ändå en bra bild hur en API till stor del fungerar.

De vanligaste Web API:s som används idag är Representational State Transfer (REST), Simple Object Access Protocol (SOAP) och Remote Procedure Call (RPC). [13]





**Figur 2. API exempel**

### 2.1.5 REST API

REST API eller Representational State Transfer är en av de vanligaste Web Api:s som används idag och används bland annat av Twitter, Google och LinkedIn. En REST API använder sig av HTTP-förfrågningar genom att använda anropen GET, POST, PUT och DELETE för att behandla data på olika sätt. Både JSON och XML format går att använda i en REST API. Ett exempel med en databas innehållande bilar skulle med hjälp av de olika anropen ge svaren nedan: [14]

1. GET: hämta en bil från databasen.
2. POST: skapa en ny bil i databasen.
3. DELETE: radera en bil i databasen.
4. PUT: uppdatera en bil i databasen.

## 2.2 Ubuntu

I examensarbetet användes Ubuntu på Windows 10 för att göra det möjligt att direkt från egen bärbara dator jobba mot uppdragsgivarens webserver. Ubuntu på Windows gör det möjligt att använda Ubuntu Terminal och med hjälp av den köra olika kommandon. Ubuntu

är ett fritt operativsystem med Linuxkärna som skapades av Mark Shuttleworth och ett litet team med utvecklare från Debian i samarbete med Canonical år 2004 med målet att skapa ett lättanvänt operativsystem som skulle vara tillgängligt för alla på samma villkor oavsett om man är ett stort företag eller en studerande. [1]

## **2.3 Unifaun**

Unifaun är den ledande tjänsten av TA-system i norra Europa. Ett TA-system är en tjänst byggd för att förenkla kommunikationen mellan transportörer och transportköpare. Förkortningen TA står för transportadministration. I sin enklaste form innebär det att du som transportör får in digitala och standardiserade transportbokningar samtidigt som dina kunder får en enklare hantering av sina transporter. [2]

### **2.3.1 Unifaun Online**

Unifaun Online är Unifauns internetbaserade system för transportadministration som hanterar transportörer och sändningar på ett och samma ställe. Beroende på vilken nivå av tjänsten som används finns det olika tillägg att använda bland annat till att boka transport, olika sorters uppföljningstjänster och tillgång till adressbok där avsändar- och mottagaradresser sparas och på så vis blir det inte något extra jobb med att skriva in adresserna i transportdokumenten. [10]

### **2.3.2 Unifaun REST API**

Unifaun REST API dokumentation är en REST API som finns tillgänglig för alla kunder som använder sig av Unifaun Online. Med hjälp av dokumentationen kan man bland annat få tillgång till fraktdokument, spårningsinformation och möjliga platser att hämta försändelser på. Med hjälp av GET, POST och DELETE-förfrågningar kan man med JSON data använda sig av olika anrop för att få tillgång till önskad information. Om felaktig eller inte tillräckligt med data finns tillgänglig i ett anrop returneras ett error meddelande och inga ändringar sker i databasen. Tillgång till hela Unifauns REST API dokumentation kan fås på adressen nedan:

<https://api.unifaun.com/rs-docs/>

Ett exempel på en POST-förfrågan där användaren matar in en ny försändelse kan bland annat se ut som exemplet på nästa sida genom att mata in data som ett JSON-objekt: [15]

**Kodexempel 3. POST-förfrågan i Unifauns REST API**

```

{
  "pdfConfig": {
    "target4XOffset": 0,
    "target2YOffset": 0,
    "target1Media": "laser-ste",
    "target1YOffset": 0,
    "target3YOffset": 0,
    "target2Media": "laser-a4",
    "target4YOffset": 0,
    "target4Media": null,
    "target3XOffset": 0,
    "target3Media": null,
    "target1XOffset": 0,
    "target2XOffset": 0
  },
  "shipment": {
    "sender": {
      "email": "tester@msn.com",
      "zipcode": "65100",
      "name": "Test Sender",
      "address1": "Testgatan 5",
      "country": "FI",
      "city": "Vasa"
    },
    "senderPartners": [{
      "id": "KK",
      "custNo": "223311"
    }],
  },
}

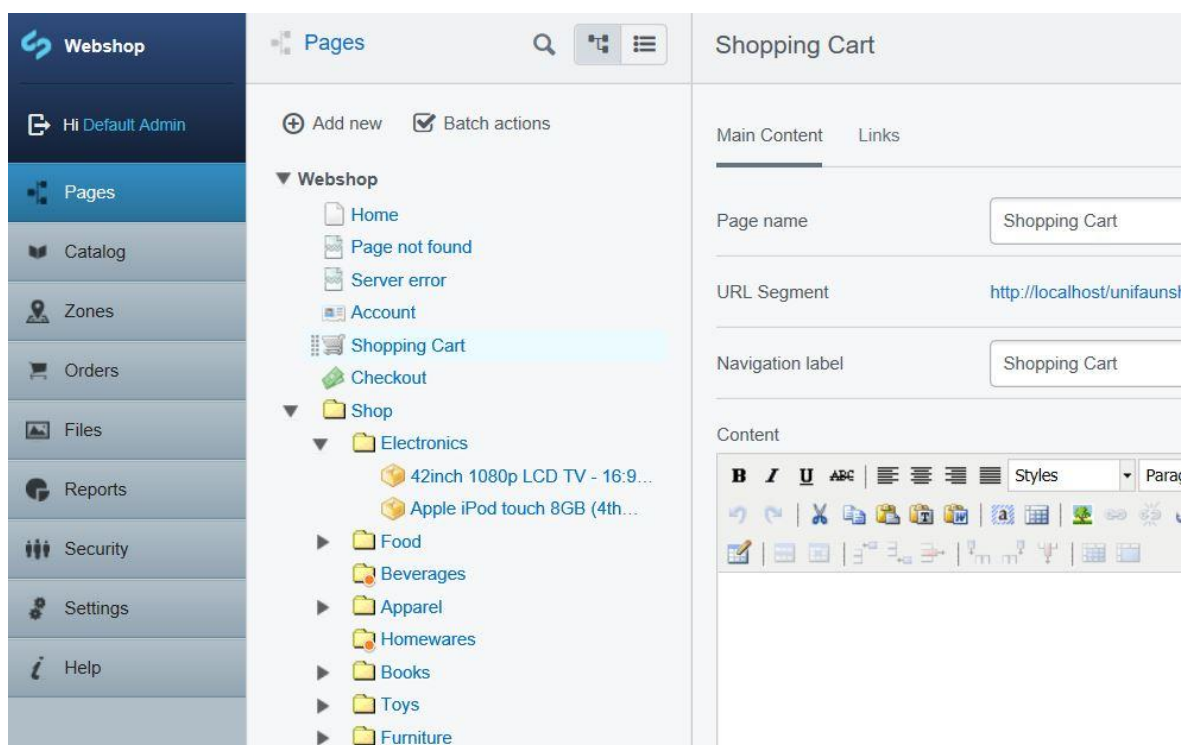
```

```
"parcels": [{
  "copies": "1",
  "weight": "5",
  "packageCode": "PC",
  "valuePerParcel": true
}],
"receiver": {
  "email": "testing@netikka.fi",
  "zipcode": "65100",
  "address1": "Testgatan 34",
  "name": "Test Receiver",
  "country": "FI",
  "city": "VASA"
},
"options": [{
  "message": "This is a testorder",
  "to": "testing@netikka.fi",
  "id": "ENOT",
  "languageCode": "FI",
  "from": "tester@msn.com",
  "sendEmail": true
}],
"service": {
  "id": "KKSTD"
}
}
```

## 2.4 Silverstripe

Silverstripe är ett Content Management System (CMS) med öppen källkod som används till att bland annat uppdatera och skapa nya webbsidor. Content Management System kan översättas till Innehållshanteringssystem. Silverstripe grundades år 2000 av Sam Minnée, Tim Copeland och Sigurd Magnusson med visionen att omforma affärer via webben. År 2006 blev Silverstripe ett innehållshanteringssystem med öppen källkod vilket gjorde att designers och utvecklare i hela världen kunde vara med och förbättra och utveckla Silverstripe. Idag har systemet laddats ner över en miljon gånger och används av allt från statliga till ideella organisationer.

Innehållshanteringssystemet är skriven i PHP och kan användas i Linux-, Apple- eller Windows-miljö. Silverstripe har även tagit fram Silverstripe Platform som gör det möjligt för klienter att stödja och hantera deras applikationer i molnet. Silverstripe Platform är ett viktigt verktyg för företag där avbrott i deras tjänster skulle orsaka stora ekonomiska förluster. Företaget har idag över 80 anställda på deras kontor i Auckland, Wellington och London samt olika samarbetspartners runt om i världen. [3]



Figur 3. Exempel på Silverstripe CMS.

## 2.5 Apache

Apache HTTP Server som oftast bara kallas Apache är utvecklad av Apache Software Foundation och är den mest använda webbservern. Idag används Apache som webbserver på över hälften av alla tjänster som använder sig av en webbserver. [6] Apache är en tjänst med öppen källkod och är en säker, pålitlig och snabb webbserver som används främst på Linuxsystem men är även tillgänglig för andra operativsystem som till exempel Windows.

En server är i allmänhet en fjärrdator som används för att leverera filer till klienter som vill komma åt informationen på servern. En webbserver kan man däremot säga är själva miljön som webbplatsen körs i. [6]

## 2.6 MariaDB

MariaDB har idag över 12 miljoner användare och är därmed en av de snabbaste växande databaserna med öppen källkod i världen och används bland annat av företag som Wikipedia, HP och booking.com. MariaDB grundades år 2009 av samma personer som grundade MySQL som ett alternativ till den populära relationsdatabasen med öppen källkod. [7]

## 2.7 Git

Git är ett distribuerat versionshanteringssystem med öppen källkod som används med snabbhet och effektivitet för att hantera allt från små till väldigt stora projekt. De viktigaste uppgifterna för ett versionshanteringssystem är att ladda upp (commit) valbar version av projektet, hämta den senaste eller en äldre version av projektet (fetch) eller synkronisera projektet med servern (push). En bra funktion med Git är att man kan skapa lokala grenar (branches) som gör det möjligt att man lokalt kan ladda upp eller hämta projektet samtidigt som andra kan jobba på andra delar av samma projekt under tiden. [8]

## 2.8 Composer

Composer är en dependency manager eller beroendehanterare på svenska för PHP som samlar alla beroenden och bibliotek som projektet behöver på ett och samma ställe. Med hjälp av Composer är det därmed lättare att hantera flera olika bibliotek samtidigt i ett projekt utan att själv vara tvungen att hålla koll på alla olika beroenden det kan innehålla. [9]

### 3 Planering

I detta kapitel beskrivs specifikationerna och det förberedande arbetet inför utförandet av examensarbetet.

#### 3.1 Specifikationer

Det viktigaste med examensarbetet var att påbörja utvecklingen av en tjänst som gör det möjligt för uppdragsgivarens kunder att genom en integration av Unifaun använda deras tjänster i webbutikerna. Inga speciella krav ställdes på själva användargränssnittet och därför användes till stor del Silverstripes egna användargränssnitt i examensarbetet.

#### 3.2 Förberedelse

Först installerades tillsammans med uppdragsgivaren de verktyg som behövdes för att utföra arbetet på egen bärbara dator. Ubuntu för Windows samt de andra verktygen som presenterats i kapitel två installerades och efter det var det bara att bekanta sig med Silverstripe med hjälp av olika typer av lektioner som finns tillgängliga på deras webbplats.

## Shopping Cart

Product	Unit Price	Quantity	Total Price (EUR)	Remove
<b>Paper Roll</b>	€12.00	<input type="text" value="1"/>	€12.00	<input type="checkbox"/>
<b>Sub-total</b>			€12.00	

**UPDATE CART**

[Continue Shopping](#) [Proceed to Checkout](#)

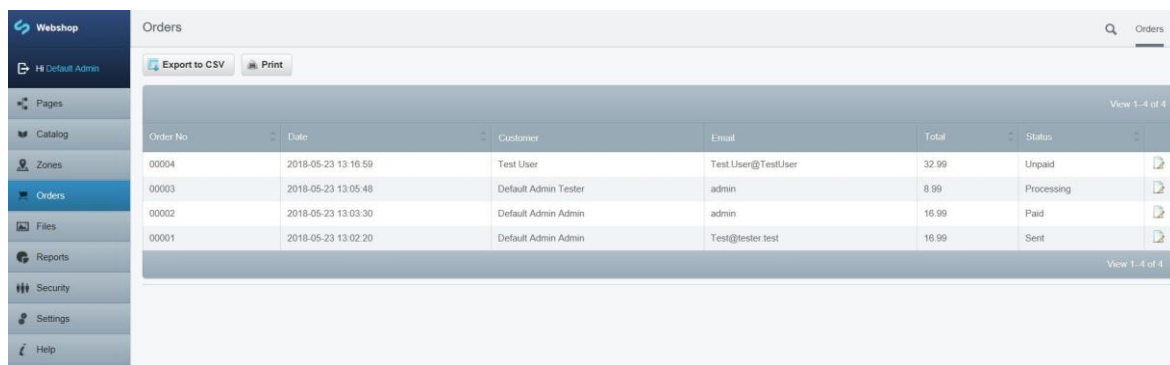
Figur 4. Exempel på kundvagnen i Silvershop.

## 4 Utförande

I detta kapitel beskrivs det hur skapandet av Silvershop och Omnipay gick till samt hur påbörjandet av integrationen till Unifaun gick till.

### 4.1 Skapande av projektet

Silvershop och Omnipay modulerna installerades som ett nytt Silverstripe projekt med hjälp av Composer och modifierades så att till exempel valutan som syns i figur 2 i föregående kapitel ändras från dollar till euro. Eftersom Silvershop använder sig av öppen källkod och ständigt är under uppbyggnad var det till exempel inte ännu möjligt att på ett enkelt sätt ändra positionen på € tecknet i versionen som användes i arbetet. I projektet skapades senare en manuell betalnings gateway som administratören kan styra över för att få in en del testbeställningar i administratörgränssnittet som kan användas för att testa integrationen till Unifaun. Administratörgränssnittet är alltså den del av projektet som bara administratören av projektet har tillgång till. Statusen för beställningen kan ändras och alla beställningar får bland annat ett beställningsnummer.



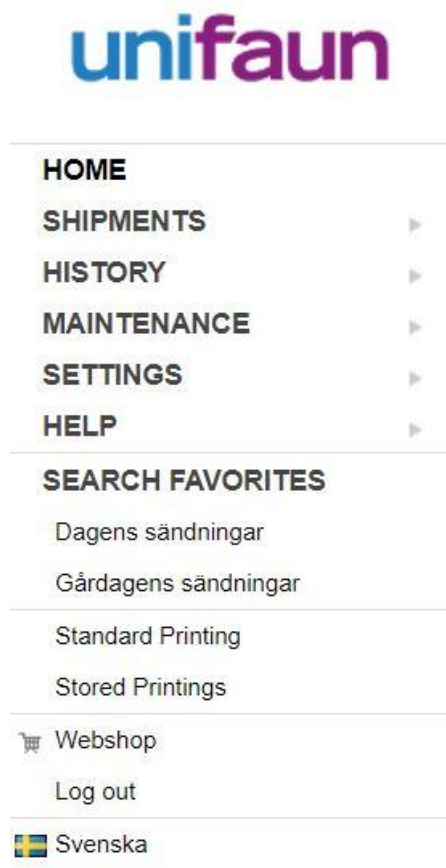
The screenshot shows the 'Orders' section of the Silvershop admin interface. On the left is a sidebar with navigation links: Webshop, Default Admin, Pages, Catalog, Zones, Orders (highlighted), Files, Reports, Security, Settings, and Help. The main area displays a table of orders with columns: Order No, Date, Customer, Email, Total, and Status. There are four test orders listed. Above the table are buttons for 'Export to CSV' and 'Print'. A search icon and the word 'Orders' are in the top right corner of the main area.

Order No	Date	Customer	Email	Total	Status
00004	2018-05-23 13:16:59	Test User	Test User@TestUser	32.99	Unpaid
00003	2018-05-23 13:05:48	Default Admin Tester	admin	8.99	Processing
00002	2018-05-23 13:03:30	Default Admin Admin	admin	16.99	Paid
00001	2018-05-23 13:02:20	Default Admin Admin	Test@tester test	16.99	Sent

**Figur 5. Testbeställningar i projektets admingränssnitt.**

För att utföra integrationen kontaktades Unifaun för att få tillgång till ett testkonto på deras tjänst Unifaun Online. I tjänsten skapades en API-nyckel för att få tillgång till deras REST-API bibliotek.





Figur 6. Användargränssnittet i Unifaun Online har en meny till vänster. (Unifaun Online, 2018)

## 4.2 Guzzle förfrågningar

Med hjälp av Guzzle skapades förfrågningar för några av de vanligaste leverantörerna som uppdragsgivarens kunder använder i webbutikerna. Förfrågningar för Posti, Matkahuolto och Kaukokiito skapades och testades sedan direkt mot uppdragsgivarens modifierade Silvershop för bästa möjliga resultat.

För att få förfrågningarna att fungera krävs autentisering med hjälp av en API-nyckel av typen Web Services (REST). Genom att fylla i Användar-Id (Developer-Id) som levererades till angiven e-post vid ansökan om ett Unifaun Online konto och kryssa i Web-Services (REST) och sedan klicka på Finish genereras en ny API-nyckel som går att använda vid förfrågningarna. Vid skapande av flera olika API-nycklar kan det vara bra att skriva en notis under Note för att ha någon sorts ordning på vilken API-nyckel som används till vilken tjänst. E-postadress och giltiga IP-adresser är andra möjliga fält som går att använda vid behov.

## New API Key

**Finish** **Cancel**

Basic Information

Status

Enabled ▾

Type

☐ Unifaun OnlineConnect
 ☒ Web Services (REST)
 ☐ Web Services (SOAP)
 ☐ XML-posting

Note

Valid IP Addresses

Email

Developer-Id

**Finish** **Cancel**

**Figur 7. Skapande av API-nyckel. (Unifaun Online, 2018)**

I Unifaun Online fanns det också en kodlista tillgänglig där viktiga koder för de olika leverantörerna och deras olika leverantörssätt kunde hittas. Kodlistan sparades undan som en Excel-fil.

Eftersom endast ett testkonto användes vid utförandet av examensarbetet kunde inte alla funktioner testas ordentligt. Förfrågningarna resulterade dock i att testbeställningar kunde köras och data sparades i Unifaun Online. Vid användning av ett riktigt Unifaun Online konto skulle verkliga förfrågningar även skicka ut ett e-postmeddelande med information och spårningsmöjligheter efter ett köp i webbutiken.

En testbeställning med en Guzzle POST-förfrågan som använder sig av Kaukokiito som leverantör och autentisering med hjälp av en API-nyckel kan se ut som koden nedan:

### Kodexempel 4. Exempel på Guzzle POST-förfrågan.

```
<?php
require_once __DIR__ . '/vendor/autoload.php';

$client = new GuzzleHttp\Client();
$response =
$client->post('https://api.unifaun.com/rs-extapi/v1/shipments', [
```

```

'auth' => [
  'API-nyckel-ID',
  'API-nyckel-Secret-ID'
],
'json' => [
  "pdfConfig" => array(
    "target4XOffset"=> 0,
    "target2YOffset"=> 0,
    "target1Media"=> "laser-ste",
    "target1YOffset"=> 0,
    "target3YOffset"=> 0,
    "target2Media"=> "laser-a4",
    "target4YOffset"=> 0,
    "target4Media"=> null,
    "target3XOffset"=> 0,
    "target3Media"=> null,
    "target1XOffset"=> 0,
    "target2XOffset"=> 0
  ),
  "shipment" => array(
    "sender" => array(
      "email"=> "tester@msn.com",
      "zipcode"=> "65100",
      "name"=> "Test Sender",
      "address1"=> "Testgatan 5",
      "country"=> "FI",
      "city"=> "VASA"
    ),
    "senderPartners"=> array(array(
      "id"=> "KK",
      "custNo"=> "223311"
    )),
    "parcels"=> array(array(
      "copies"=> "1",
      "weight"=> "5",
      "packageCode"=> "PC",
      "valuePerParcel"=> true
    )),
    "receiver" => array(
      "email"=> "testing@netikka.fi",
      "zipcode"=> "65100",
      "name"=> "Test Receiver",
      "address1"=> "Testgatan 34",
      "country"=> "FI",
      "city"=> "VASA"
    ),
    "service"=> array(
      "id"=> "KKSTD"
    ),
    "options"=> array(array(
      "message"=> "Testing testing",
      "to"=> "testing@netikka.fi",
      "id"=> "ENOT",
      "languageCode"=> "FI",
      "from"=> "tester@msn.com",
      "sendEmail"=> true
    ))
  )
];

```

```
echo $response-&gtgetStatusCode();  
echo $response->getBody();
```

I koden används en standardmall för pdf dokument och i "shipment" delen av koden fyller man bland annat i info om sändaren och mottagaren. Ur kodlistan med leverantörer hittas Kaukokiitos "service" id "KKSTD" och "senderPartners" id "KK" som kundnummer eller "custNo" fungerar en sexsiffrig kod som i detta exempel valdes slumpmässigt eftersom detta bara är en testbeställning men olika företagare med webbutiker har tillgång till ett eget kundnummer att använda.

Under "parcels" finns innehåll om själva paketet som sänds och under "options" med id "ENOT" som är id för "E-mail pre-notification" finns information om vart e-postmeddelandet med bland annat spåringsmöjligheterna ska skickas.

Som svar på förfrågan fås HTTP statuskod 201 för att bekräfta att en ny "shipment" har skapats i Unifaun Online samt en JSON-sträng med data. Svaret ger bland annat tillgång till pdf dokument som innehåller fraktsedlar som kan printas ut och klistras på paketet när det skickas iväg. På nästa sida kan JSON-svaret ses:

**Kodexempel 5. JSON-svar på HTTP-förfrågan.**

```
[  
  {  
    "href": "https://api.unifaun.com/rs-extapi/v1/shipments/11021459",  
    "id": "11021459",  
    "status": "PRINTED",  
    "shipmentNo": "136655613198",  
    "orderNo": null,  
    "reference": null,  
    "serviceId": "KKSTD",  
    "parcelCount": 1,  
    "sndName": "Test Sender",  
    "sndZipcode": "65100",  
    "sndCity": "VASA",  
    "sndCountry": "FI",  
    "rcvName": "Test Receiver",  
    "rcvZipcode": "65100",  
    "rcvCity": "VASA",  
    "rcvCountry": "FI",  
    "created": "2018-09-04T08:47:37.342+0000",  
    "changed": "2018-09-04T08:47:37.342+0000",  
    "shipDate": "2018-09-04T08:47:37.000+0000",  
    "returnShipment": false,  
    "normalShipment": true,  
    "consolidated": false,  
  }  
]
```

```
"parcels": [  
  {  
    "parcelNo": "373325381494456186",  
    "returnParcelNo": null,  
    "reference": null  
  },  
  {  
    "href":  
"https://api.unifaun.com/rs-extapi/v1/shipments/11021459/pdfs/56688042",  
    "id": "56688042",  
    "description": "Label",  
    "pdf": null  
  },  
  {  
    "href":  
"https://api.unifaun.com/rs-extapi/v1/shipments/11021459/pdfs/56688043",  
    "id": "56688043",  
    "description": "Doc",  
    "pdf": null  
  },  
  {  
    "previousPdfs": null  
  }  
]
```

## 5 Resultat och diskussion

Resultatet av examensarbetet blev ett första steg på integrationen av Unifaun till uppdragsgivarens Silvershopmodul. Koden som arbetats fram och förfrågningarna som körs med Guzzle kan användas för att registrera testbeställningar och kan användas för att vidareutveckla produkten.

Nästa steg är att med hjälp av kunder som är intresserade av en integration med Unifaun vidareutveckla en version som fungerar bra med deras webbutik.

Själva arbetet med att få fram en integration med Unifaun var väldigt intressant men även ganska tidskrävande då mycket av verktygen och tjänsterna som användes var nya för mig. Projektet med att få en fullständigt fungerande version att erbjuda direkt åt uppdragsgivarens kunder visade sig vara krävande men med hjälp av detta examensarbete finns det dock en bra grund att jobba vidare på för att utveckla en färdig produkt.

Silverstripe är ett stort innehållshanteringssystem och med examensarbetet har jag lärt mig grunderna i hur det fungerar men för att få en bättre bild av hur det fungerar behöver man jobba med flera projekt som använder sig av det.

Examensarbetet var mycket lärorikt och med hjälp av det känns det nu som jag har en större kunskap inom många områden inom informationsteknik.

Förhoppningsvis gör detta examensarbete det möjligt att i framtiden ta fram en version av integrationen som kommer att användas av många av uppdragsgivarens kunder.

## 6 Källförteckning

- [1] Ubuntu, *The story of Ubuntu*. (u.å.) [Online]  
<https://www.ubuntu.com/about> [hämtat 16.5.2018].
- [2] Unifaun, *TA för transportörer*. (u.å.) [Online]  
<https://www.unifaun.com/sv/transportor/ta-for-transportorer/> [hämtat 11.5.2018].
- [3] Silverstripe, *who we are*. (u.å.) [Online]  
<https://www.silverstripe.com/who-we-are/history/> [hämtat 14.5.2018].
- [4] PHP, *PHP-guiden*. (u.å.) [Online]  
<http://www.webbdesignguiden.se/php-guiden/> [hämtat 14.5.2018].
- [5] PHP, *PHP manual*. (u.å.) [Online]  
<http://php.net/manual/en/intro-whatcando.php> [hämtat 14.5.2018].
- [6] B. Mitchell, 2017. *Definition of apache*. [Online]  
<https://www.lifewire.com/definition-of-apache-816509> [hämtat 15.5.2018].
- [7] MariaDB, 2018. *About us*. [Online]  
<https://mariadb.com/about-us> [hämtat 15.5.2018].
- [8] Versionshantering, *Om versionshantering*. (u.å.) [Online]  
<https://www.ida.liu.se/~TDP001/material/tutorials/vc/> [hämtat 16.5.2018].
- [9] Composer, 2013. *What is PHP Composer*. [Online]  
<https://www.culttt.com/2013/01/07/what-is-php-composer/> [hämtat 16.5.2018].
- [10] Unifaun, *Unifaun-online*. (u.å.) [Online]  
<https://www.unifaun.com/sv/transportkopare/unifaun-online/> [hämtat 24.5.2018].
- [11] Guzzle, 2015. *Guzzle documentation*. [Online]  
<http://docs.guzzlephp.org/en/stable/> [hämtat 2.9.2018].
- [12] JSON, *Json - Introduction*. (u.å.) [Online]  
[https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp) [hämtat 2.9.2018].
- [13] API, 2018. *What is an API?* [Online]  
<https://www.howtogeek.com/343877/what-is-an-api/> [hämtat 3.9.2018].
- [14] REST API, 2016. *RESTful API*. [Online]  
<https://searchmicroservices.techtarget.com/definition/RESTful-API>  
[hämtat 3.9.2018].
- [15] Unifaun REST API, *REST API Documentation*. (u.å.) [Online]  
<https://api.unifaun.com/rs-docs/##introduction> [hämtat 3.9.2018].